Elliptic Curve Digital Signature Algorithm (ECDSA)
Departemen Teknik Informatika ITB

Andy Triwinarko

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if18017@students.if.itb.ac.id, andytreeweenarko@yahoo.com

Abstrak

Kriptografi kurva eliptik termasuk kedalam sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik. Tidak seperti permasalahan matematis logaritma diskrit (*Discrete Logarithm Problem, DLP*) dan pemfaktoran bilangan bulat (*Integer Factorization Problem, IFP*), tidak ada algoritma waktu subeksponensial yang diketahui untuk memecahkan permasalahan matematis logaritma diskrit kurva eliptik (*Elliptic Curve Discrete Logarith Problem, ECDLP*). Karena alasan tersebut, algoritma kriptografi kurva eliptik mempunyai keuntungan jika dibandingkan dengan algoritma kriptografi kunci publik lainnya yaitu dalam hal ukuran panjang kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama. Ada tiga protokol ECDLP yang diketahui saat ini yaitu *Elliptic Curve Digital Signature Algorithm (ECDSA)*, *Elliptic Curve Diffie Hellman (ECDH*), dan *Elliptic Curve ElGamal (ECElgamal)*. Jurnal ini membahas tentang *ECDSA* dan pengimplementasiannya, serta pembahasan tingkat keamanan dan performansinya.

***Kata kunci**: sistem kriptografi kunci publik, kriptografi kurva eliptik, ECDSA, tingkat keamanan, dan performansi*

1. Pendahuluan

Ketika saling berkomunikasi dengan pihak lain melalui dunia maya, terkadang diperlukan proses pertukaran dokumen elektronis (*file*). Hal ini memerlukan adanya suatu mekanisme untuk menjamin keaslian (otentikasi) dokumen elektronis yang bersangkutan. Metode yang sering digunakan untuk mengatasi permasalahan di atas adalah dengan cara menambahkan (meng-*embedded*) tanda tangan digital pada dokumen elektronis tersebut. Tanda tangan pada dokumen elektronis ini disebut tanda tangan digital (*digital signature*). Dengan tanda tangan digital, maka integritas data dapat dijamin, disamping itu ia juga digunakan untuk membuktikan asal pesan (keabsahan pengirim dan anti-penyanggahan).

Sistem kriptografi yang cocok digunakan untuk tanda tangan digital adalah sistem kriptografi kunci-publik. Hal ini disebabkan karena skema tanda tangan digital berbasis sistem kunci-publik dapat menyelesaikan masalah non-repudiation (baik penerima dan pengirim pesan mempunyai pasangan kunci masing-masing). Sistem kriptografi kunci publik mempunyai tingkat keamanan (security level) yang sebanding dengan jumlah kunci (bit) yang dipakai, atau dengan kata lain semakin panjang ukuran kunci maka semakin tinggi pula tingkat keamanannya. Secara umum permasalahan tersebut tidak terlalu signifikan bila diimplementasikan di PC (*Personal Computer*), tetapi akan menjadi suatu masalah yang besar untuk peralatan dengan kapasitas memori dan daya untuk proses yang sangat terbatas seperti *smart cards, hand phone, PDA, tablet PC*, dan peralatan *wireless* lainnya yang berkembang sangat pesat akhir-akhir ini. Sehingga diperlukan sebuah algoritma kriptografi kunci yang mempunyai tingkat keamanan tinggi (high security level), tetapi menggunakan ukuran kunci yang relatif kecil.

Untuk mengatasi permasalahan tersebut, pada tahun 1985, Victor Miller dan N. Koblitz menawarkan solusi yang berupa teknik kriptografi berdasarkan pendekatan matematika dengan menggunakan kurva eliptik, yang lebih dikenal dengan nama kriptografi kurva eliptik (*Elliptics Curve Cryptography*). Saat ini, kriptografi kurva eliptik yang ada menggunakan pendekatan logaritma diskrit, yang biasa disebut dengan ECDLP (*Elliptic Curve Discret Logarithm Problem*). Ada tiga algoritma dalam ECDLP yaitu: ECDSA (*Elliptic Curve-DSA*), ECDH (*Elliptic Curve Diffie Hellman*), dan ECElGamal. ECDSA merupakan analog dari *Digital Signature Algorithm (DSA)* yang diterapkan pada kurva eliptik.

2. Tanda Tangan Digital

Tanda tangan digital dengan menggunakan fungsi *hash* satu arah (*one way hash function*) secara

umum mempunyai tiga macam proses utama, yaitu : pembangkitan pasangan kunci, pemberian tanda tangan digital (*signing*), dan verifikasi terhadap keabsahan tanda tangan digital tersebut (*verifying*).

***Signing**, p*esan yang hendak dikirim diubah terlebih dahulu menjadi bentuk yang ringkas yang disebut *message digest*. *Message digest* (*MD*) diperoleh dengan cara mentransformasikan pesan *M* menggunakan fungsi *hash* satu-arah (*one-way*) *H*,

$$MD = H(M) \quad (1)$$

Pesan yang sudah diubah menjadi *message digest* oleh fungsi *hash* tidak dapat dikembalikan lagi menjadi bentuk semula walaupun digunakan algoritma dan kunci yang sama (itulah sebabnya dinamakan fungsi *hash* satu-arah). Sembarang pesan yang berukuran apapun diubah oleh fungsi *hash* menjadi *message digest* yang berukuran tetap (umumnya 128 bit). Selanjutnya, *message digest* MD dienkripsikan dengan algoritma kunci-publik menggunakan kunci rahasia (SK) pengirim menjadi tanda tangan tanda tangan *S*,

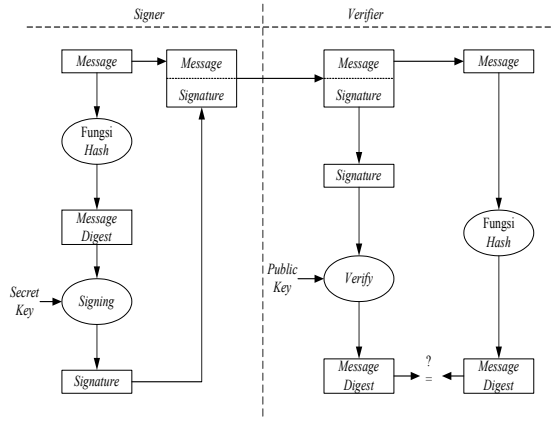
$$S = E_{SK}(MD) \quad (2)$$

Pesan *M* disambung (*append*) dengan tanda tangan tanda tangan *S*, lalu keduanya dikirim melalui saluran komunikasi. Dalam hal ini, kita katakan bahwa pesan *M* sudah ditandatangani oleh pengirim dengan tanda tangan digital *S*.

Verifying, Pesan *M* dan tanda tangan digital *S* yang dikirim melalui saluran komunikasi akan diterima oleh pihak penerima. Di tempat penerima, pesan diverifikasi untuk dibuktikan keotentikannya dengan cara berikut : Tanda tangan digital *S* didekripsi dengan menggunakan kunci publik (*PK*) pengirim pesan, menghasilkan *message digest* semula, *MD*, sebagai berikut:

$$MD = D_{PK}(S) \quad (3)$$

Pengirim kemudian mengubah pesan *M* menjadi *message digest MD'* menggunakan fungsi *hash* satu-arah yang sama dengan fungsi *hash* yang digunakan oleh pengirim. Jika *MD' = MD*, berarti pesan yang diterima otentik dan berasal dari pengirim yang benar.



Proses pembuktian keotentikan tanda tangan digital ini dijelaskan sebagai berikut:

1. Apabila pesan *M* yang diterima sudah berubah, maka *MD'* yang dihasilkan dari fungsi *hash* berbeda dengan *MD* semula. Hal ini berarti bahwa pesan sudah tidak asli lagi (*data integrity*).
2. Apabila pesan *M* tidak berasal dari orang yang sebenarnya, maka *message digest MD* yang dihasilkan dari persamaan 3 berbeda dengan *message digest MD'* yang dihasilkan pada proses verifikasi (hal ini karena kunci publik yang digunakan oleh penerima pesan tidak berkoresponden dengan kunci rahasia pengirim). Bila *MD = MD'*, ini berarti pesan yang diterima adalah pesan yang asli (*message authentication*) dan orang yang mengirim adalah orang yang sebenarnya (*user authentication*). Karena proses *signing* menggunakan kunci rahasia pengirim maka pengirim pesan tidak dapat menyangkal aktivitas yang telah dilakukannya (*non-repudiation*).

3. Bidang Terbatas

Bidang terbatas (*finite field)* atau yang biasa disebut dengan *Galois Field* (GF) adalah bidang yang hanya memiliki elemen bilangan yang terbatas. Derajat (*order*) dari *finite field* adalah banyaknya elemen yang ada di dalam bidang. Jika *q* adalah pangkat prima (*prime power*), maka hanya ada satu bidang terbatas dengan derajat *q*. Bidang tersebut dilambangkan dengan F_q atau *GF(q)*. Banyak cara untuk merepresentasikan elemen dari F_q, jika $q=p^m$, dimana *p* adalah bilangan prima dan *m* adalah bilangan integer positif, maka *p* disebut sebagai karakteristik dari F_q dan *m* disebut sebagai derajat perluasan (*extension degree*) dari F_q. Bidang terbatas yang digunakan dalam kriptografi adalah *q=p*, dimana *p* adalah bilangan prima ganjil, yang dilambangkan dengan F_p (*odd prime*), dan $q=2^m$, dimana m adalah integer lebih besar dari satu, yang dilambangkan dengan F_2^m (*characteristic two or even*).

3.1 Bidang Terbatas F_p

Bidang Terbatas F_p merupakan sebuah bidang yang beranggotakan bilangan integer {0,1,...,*p-1*}, dan *p* merupakan bilangan prima, setiap perhitungan dikalkulasikan dengan modulo *p* agar hasilnya tetap berada dalam daerah F_p. Operasi yang berlaku dalam bidang terbatas F_p adalah:

1. Penjumlahan (*Addition*), jika $a,b \in F_p$, maka $a + b = r$, dimana *r* adalah sisa pembagian $a + b$ dengan bilangan prima *p*, $0 \le r \le p-1$. penjumlahan seperti ini disebut penjumlahan modulo *p* (mod *p*).
2. Perkalian (*Multiplication*), jika $a,b \in F_p$, maka $a \bullet b = s$, dimana *s* adalah sisa pembagian $a \bullet b$

dengan bilangan prima p, $0 \le s \le p-1$. perkalian seperti ini disebut perkalian modulo p (mod p).

3.2 Bidang Terbatas F_2^m

Bidang terbatas F_2^m biasa disebut dengan bidang terbatas biner (*binary finite field*), dapat dipandang sebagai ruang vektor berdimensi m pada F_2. Karena itu ada himpunan yang beranggotakan m elemen $\{\alpha_0,\alpha_1,...\alpha_{m-1}\}$di dalam F_2^m sedemikian rupa sehingga setiap a$\in F_2^m$ dapat ditulis secara unik ke dalam bentuk:

$a = a_0\alpha_0+a_1\alpha_1+...+a_{m-1}\alpha_{m-1}$, untuk $ai \in \{0,1\}$ (4)

Salah satu cara untuk merepresentasikan elemen-elemen pada F_2^m adalah dengan representasi basis polinomial. Pada representasi basis polinomial elemen pada F_2^m merupakan polinomial dengan derajat lebih kecil dari m, dengan koefisien bilangan 0 atau 1.

$\{a_{m-1}x^{m-1} + ... + a_2x^2 + a_1x^1 + a_0x^0 \mid a_i : 0,1\}$ (5)

Operasi yang berlaku dalam bidang terbatas F_2^m representasi basis polinomial:

1. Penjumlahan (*Addition*), $(a_{m-1}...a_1a_0) + (b_{m-1}..b_1b_0) = (c_{m-1}...c_1c_0)$ dimana $c_i = a_i + b_i$. Operasi penjumlahan dapat menggunakan deretan komponen $(a_{m-1}...a_1a_0)$ yang di-XOR-kan dengan $(b_{m-1}..b_1b_0)$.
2. Perkalian (Multiplication), $(a_{m-1}...a_1a_0) \bullet (b_{m-1}..b_1b_0) = (r_{m-1}...r_1r_0)$ dimana $r_{m-1}x^{m-1} + ... + r_1x + r_0$ adalah sisa dari pembagian $(a_{m-1}x^{m-1} + ... + a_1x + a_0) . (b_{m-1}x^{m-1} + ... + b_1x + b_0)$ dibagi dengan polinomial $f(x)$ pada F_2 (setiap koefisien polinomial di reduksi ke modulo 2).

4. Kurva Eliptik Pada Bidang Terbatas

Ada beberapa cara untuk mendefinisikan persamaan kurva eliptik bergantung kepada bidang terbatas yang digunakan apakah F_p atau F_2^m. Persamaan Weierstrass yang digunakan untuk kedua bidang terbatas tersebut berbeda.

4.1 Kurva Eliptik Pada Bidang Terbatas F_p

Misalkan $p > 3$ adalah bilangan prima ganjil, dan $a,b \in F_p$ memenuhi

$$4a^3 + 27b^2 \neq 0 \ (mod\ p) \quad (6)$$

maka sebuah kurva eliptik E(F_p) pada F_p merupakan himpunan titik-titik $P(x,y)$, dimana $x,y \in F_p$,yang memenuhi persamaan :

$$y^2 = x^3 + ax + b, \quad (7)$$

dan sebuah titik khusus φ(∞,∞) yang merupakan titik tak hingga. Operasi penjumlahan pada $E(F_p)$ didefinisikan sebagai berikut :

1. $P + \varphi = \varphi + P = P$ untuk setiap $P \in E(F_p)$

Jika $P(x.y) \in E(F_p)$, maka $(x,y) + (x,-y) = \varphi$ (titik $(x,-y) \in E(F_p)$ dinotasikan sebagai -P, disebut sebagai negatif dari P)

2. Misalkan $P(x_1,y_1) \in E(F_p)$, $Q(x_2,y_2) \in E(F_p)$, dan $P \neq \pm Q$, maka $P + Q = (x_3,y_3)$ dimana :

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \quad (8)$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \quad (9)$$

3. Misalkan $P(x_1,y_1) \in E(F_p)$, maka $P + P = 2P = (x_3,y_3)$, dimana :

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \quad (10)$$

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \quad (11)$$

Operasi di atas disebut dengan penggandaan titik (*doubling a point*)

Kehebatan dari operasi penjumlahan pada kurva eliptik adalah jika menjumlahkan dua buah titik yang merupakan elemen dari kelompok kurva eliptik, maka hasil penjumlahannya adalah titik lain yang juga merupakan elemen dari kelompok kurva eliptik tersebut.

4.2 Kurva Eliptik Pada Bidang Terbatas F_2^m

Sebuah kurva eliptik E pada F_2^m didefinisikan sebagai sebagai sebuah persamaan dalam bentuk :

$$y^2 + xy = x^3 + ax^2 + b \ , \quad (12)$$

dimana $a,b \in F_2^m$, dan $b \neq 0$. Set $E(F_2^m)$ terdiri dari seluruh titik (x,y), $x \in F_2^m$, $y \in F_2^m$ yang memenuhi persamaan kurva eliptik tersebut, bersamaan dengan titik khusus $\varphi(\infty,\infty)$ yang disebut titik tak hingga (*point at infinity*).

Sebagaimana kurva-kurva eliptik pada *Fp*, ada aturan-aturan untuk menjumlahkan titik–titik pada kurva eliptik $E(F_2^m)$ untuk mendapatkan sebuah titik ketiga kurva eliptik. Rumus aljabar untuk menjumlahkan dua titik dan menggandakan dua titik adalah sebagai berikut.

1. $P + \varphi = \varphi + P = P$ untuk seluruh $P \in E(F_2^m)$.

Jika $P = (x,y) \in E(F_2^m)$, kemudian $(x,y) + (x, x+y) = \varphi$. (Titik $(x,x+y)$ dinotasikan dengan $-P$, dan disebut negatif P.

2. Misalkan $P = (x_1,y_1) \in E(F_2^m)$ dan $Q = (x_2,y_2) \in E(F_2^m)$, dimana $P \neq \pm Q$. Kemudian $P + Q = (x_3,y_3)$, dimana

$$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + \alpha \quad (13)$$

$$y_3 = \left(\frac{y_1 + y_2}{x_1 + x_2} \right)(x_1 + x_3) + x_3 + y_1. \quad (14)$$

3. Penggandaan titik (*Point doubling*) Misalkan $P = (x_1,y_1) \in E(F_2^m)$, kemudian $2P = (x_3,y_3)$, dimana :

$$x_3 = x_1^{\,2} + \frac{b}{x_1^{\,2}} \quad (15)$$

$$y_3 = x_1^{\,2} + \left(x_1 + \frac{y_1}{x_1} \right) x_3 + x_3. \quad (16)$$

5. ECDSA

Dalam protokol ECDSA, pihak yang akan melakukan tanda tangan digital, mempunyai parameter domain kurva eliptik berupa $D = \{q,FR,a,b,G,n,h\}$ dan pasangan kunci kunci rahasia d_A dan kunci publik Q_A. Kemudian pihak yang akan melakukan verifikasi terhadap tanda tangan, memiliki salinan dokumen D yang otentik dan kunci publik Q_A. Proses-proses yang terjadi adalah sebagai berikut :

Key Generation

1. Memilih sebuah bilangan bulat random d_A, yang nilainya diantara $[1,n-1]$
2. Menghitung $Q_A = d_A \bullet G = (x_1,y_1)$
3. Kunci rahasia = d_A, dan kunci publik = Q_A.

Signing

1. Memilih sebuah bilangan bulat random k, yang nilainya diantara $[1,n-1]$.
2. Menghitung $Q_A = k \bullet G = (x_1,y_1)$ dan $r = x_1 \bmod n$, jika $r = 0$, maka kembali ke langkah 1.
3. Menghitung $k^{-1} \bmod n$
4. Menghitung $e = Hash(m)$
5. Menghitung $s = k^{-1} \{e + d_A \bullet r\} \bmod n$

tanda tangan Alice untuk *message m* adalah (r,s)

Verifying

1. Memverifikasi bahwa r dan s adalah bilangan bulat yang antara $[1,n-1]$
2. Menghitung $e = Hash\ (m)$
3. Menghitung $w = s^{-1} \bmod n$
4. Menghitung $u_1 = ew \bmod n$ dan $u_2 = rw \bmod n$
5. Menghitung $u_1 \bullet G + u_2 \bullet Q_A = (x_1,y_1)$
6. Menghitung $v = x_1 \bmod n$
7. Menerima tanda tangan jika dan hanya jika $v = r$

6. Implementasi

Untuk membangun sebuah kriptosistem kurva eliptik ada tiga hal yang harus diperhatikan, yaitu :

1. Pemilihan bidang terbatas F_q dan representasi elemen dari F_q, implementasi yang dipilih : F_p dengan representasi elemen berupa bilangan bulat yang sangat besar.
2. Pemilihan Kurva Eliptik E pada F_q, tidak semua kurva eliptik 'aman' digunakan untuk kriptografi. Menurut [9] , syarat yang harus dipenuhi adalah :
   a. Jumlah titik pada kurva E atau derajat kurva E, $\#E(F_q)$, harus dapat dibagi oleh sebuah bilangan prima n yang cukup besar.
   b. $\#E(F_q) \neq q$
   c. n tidak membagi q^k-1 untuk semua $1 \leq k \leq 20$

Implementasi yang dipilih : kurva eliptik yang dibangkitkan dengan cara metoda perkalian komplek (*Complex Multiplication Method*).

3. Penentuan protokol kurva eliptik, implementasi yang dipilih : protokol ECDSA.

Perangkat lunak yang diimplementasikan diberi nama **EDiS** (***E**lliptic Curve **D**igital **S**ignature*), disamping itu juga dikembangkan penandatanganan dokumen elektronis dengan menggunakan sistem kriptografi kunci publik lainnya yaitu penandatanganan dokumen menggunakan algoritma RSA. Kemudian kedua algoritma ini diperbandingkan tingkat keamanan dan performansinya.

7. Tingkat Keamanan

Yang dimaksud dengan tingkat keamanan pada sistem kriptografi kunci publik adalah berapa waktu yang diperlukan untuk memecahkan suatu kunci rahasia berdasarkan persamaan matematis yang dimiliki oleh algoritma kriptografinya. RSA termasuk ke dalam persamaan matematis *Integer Factorization Problem* (IFP) sedangkan ECDSA termasuk ke dalam *Elliptic Curve Discrete Logarithm Problem* (ECDLP). Tingkat keamanan dihitung berdasarkan panjang kunci dari masing-masing algoritma kriptografi, parameter kunci RSA yang digunakan adalah panjang bit n , yaitu perkalian antara faktor prima p dan q, sedangkan untuk ECDSA parameter kunci yang digunakan juga panjang bit n, tetapi merupakan orde dari titik basis yang digunakan dalam persamaan kurva eliptik.

Untuk memecahkan persamaan matematis tersebut harus digunakan *software* dan *hardware* yang terbaik. Menurut [1] algoritma terbaik yang diketahui untuk menyelesaikan IFP pada RSA adalah algoritma *General Purposed Number Field Sieve* yang memiliki kompleksitas algoritma $O = exp\ [1,923\ (ln\ n)^{1/3}\ (ln\ ln\ n)^{2/3}]$, sedangkan untuk menyelesaikan ECDLP pada ECDSA adalah *Pollard Rho Method Attacks* yang memiliki $O = 2^{\,n/2}$. Jika diasumsikan *hardware* yang digunakan mampu menjalankan 1000000 instruksi per detik (1 MIPS (*Million Instruction per Second*)) maka akan dihitung tingkat keamanan kunci ECDSAsebagai berikut : Misalkan untuk n = 149 bit , maka tingkat keamanan dihitung sebagai berikut :

MIPS = $2^{149/2}$ / 1000000.3600.24.365 = 598981035 MIPS years

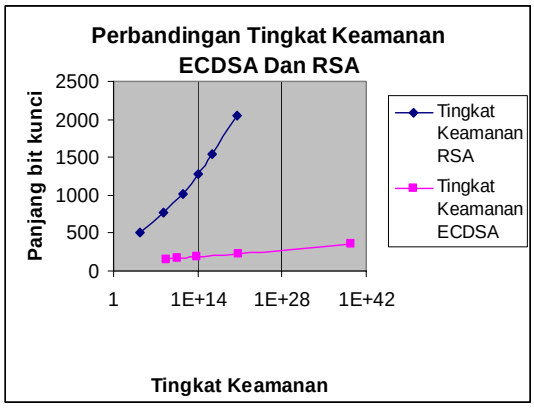
Dengan cara yang sama dihitung tingkat keamanan untuk kunci dengan panjang bit n yang berbeda-beda sehingga diperoleh tabel hubungan panjang kunci ECDSA dengan tingkat keamanannya sebagai berikut :

No	Ukuran n (bit)	Tingkat Keamanan (MIPS year) (≈)
1	149	6.10^{8}
2	160	$3{,}8.10^{10}$
3	183	$7{,}9.10^{13}$
4	229	$6{,}6.10^{20}$
5	353	3.10^{39}
6	512	$3{,}7.10^{63}$
7	768	$1{,}2.10^{102}$
8	1024	$4{,}3.10^{140}$
9	1280	$1{,}4.10^{179}$
10	1536	$4{,}9\ .10^{217}$
11	2048	$5{,}7.10^{294}$

Sedangkan untuk hubungan panjang kunci RSA dan tingkat keamanannya menurut [13] dapat dilihat dalam tabel berikut ini:

No	Ukuran n (bit)	Tingkat Keamanan (MIPS year)
1	512	30000
2	768	2.10^{8}
3	1024	3.10^{11}
4	1280	1.10^{14}
5	1536	3.10^{16}
6	2048	3.10^{20}

Hubungan antara tingkat keamanan RSA dan ECDSA dapat dilihat pada grafik berikut ini :



8. Performansi

Untuk membahas tingkat performansi dari ECDSA maupun RSA ada tiga kriteria yang menjadi pertimbangan yaitu :

Ukuran pajang kunci, kunci publik RSA adalah pasangan (n,e), dimana n adalah modulo sedangkan e adalah kunci publik. Jika sistem kriptografi RSA yang dibangun 1024 bit, maka tentunya n juga mempunyai panjang 1024 bit dan kunci publik yang digunakan adalah $e = 2^{16} + 1 = 65537$. Ukuran kunci publik RSA yang diperlukan adalah 128 bytes untuk modulo dan 3 bytes untuk kunci publik, sehingga totalnya adalah 131 bytes. Sedangkan jika sistem kriptografi ECDSA menggunakan 160 bit, maka panjang kunci publik ECDSA adalah sebuah titik pada kurva $Q(x,y)$ yang masing masing elemennya , x dan y, juga mempunyai panjang 160 bit. Sehingga total ukuran kunci publik ECDSA adalah 160 bit = 40 bytes, yang jauh lebih kecil jika dibandingkan dengan RSA.

Ukuran panjang tanda tangan digital, panjang tanda tangan digital ECDSA = 320 bit (2 x 160 bit, tanda tangan merupakan pasangan r dan s yang masing-masing panjangnya 160 bit), atau 40 bytes. Sedangkan ukuran tanda tangan digital RSA adalah 1020 bit ≈ 1024 bit = 128 bytes. Sehingga total ukuran tanda tangan digital ECDSA jauh lebih kecil dari RSA.

Kecepatan proses *signing* dan *verifying*, waktu yang diperlukan untuk proses *signing* dan *verifying* dapat dilihat pada tabel berikut ini :

Proses	ECDSA	RSA
Operasi Kunci Rahasia (pembangkitan tanda tangan digital)	Cepat	Lambat
Operasi Kunci Publik (verifikasi tanda tangan digital)	Lambat	Cepat

Berdasarkan tabel di atas ECDSA baik digunakan untuk proses yang banyak menggunakan pembangkitan tanda tangan digital, misalnya digunakan oleh orang yang sering menggunakan *webmail* karena di setiap suratnya, dia harus selalu menandatanganinya. Sebaliknya RSA baik digunakan untuk proses yang sering melakukan verifikasi tanda tangan digital, misalnya pihak CA (Certification Authority) yang hanya menadatangani sertifikat kunci publik sekali saja tetapi sertifikat tersebut nantinya akan sering diverifikasi orang lain.

9. Kesimpulan

Kesimpulan yang dapat ditarik sehubungan dengan tingkat keamanan dan performansi dari algoritma kriptografi ECDSA maupun RSA adalah :

1. ECDSA dengan panjang kunci 160 bit mempunyai tingkat keamanan yang relatif sama dengan RSA dengan panjang kunci 1024 bit. Jadi algoritma kriptografi kurva eliptik mempunyai keuntungan berupa ukuran panjang kunci yang lebih kecil jika dibandingkan dengan algoritma kunci publik lainnya (RSA) tetapi sudah memiliki tingkat keamanan yang relatif sama., sehingga algoritma kriptografi kurva eliptik cocok untuk diimplementasikan pada peralatan

perangkat keras yang memiliki daya dan memori yang terbatas

2. Dari kriteria ukuran panjang tanda tangan digital, algoritma kriptografi kurva eliptik memiliki performansi yang lebih baik karena menghasilkan tanda tangan digital yang mempunyai ukuran lebih kecil. Sedangkan dari kriteria kecepatan proses *signing* dan *verifying,* performansi kriptografi kurva eliptik akan lebih baik jika proses *signing* lebih sering dilakukan. Sebaliknya performansi kriptografi RSA akan lebih baik jika proses *verifying* lebih sering dilakukan.

10. Saran

Saran untuk pengembangan di masa mendatang :

1. Kriptografi kurva eliptik memiliki dua buah bidang terbatas yaitu F_p dan F_2^m, dan yang diimplementasikan adalah bidang terbatas F_p, saran penulis adalah bidang terbatas F_2^m juga diimplementasikan, kemudian dicoba untuk dibandingkan bagaimana performansi keduanya.
2. Algoritma kriptografi kurva eliptik lain yang dapat digunakan untuk penandatanganan dokumen elektronis adalah ECElGamal, saran penulis adalah algoritma kriptografi ECElGamal tersebut juga diimplementasikan untuk kemudian diketahui bagaimana performansinya jika dibandingkan dengan algoritma kriptografi kurva eliptik ECDSA.

11. Daftar Pustaka

1. A Certicom Whitepaper. (2000). *The Elliptic Curve Cryptosystem.* http://www.certicom.com, Agustus 2004.
2. A Certicom Whitepaper.(2000). *The Elliptic Curve Cryptosystemt, Remarks on The Security of The Elliptic Curve Cryptosystem.* http://www.certicom.com, Desember 2004
3. A DeviceForge Article and Whitepapers. (2004). *An Intro to Elliptic Curve Cryptography*. http://www.deviceforge.com, Oktober 2004.
4. IEEE 1363. (2000). *Standard Specifications for Public-Key Cryptography.* http://grouper.ieee.org/groups/1363/index.html Januari 2005
5. Johnson, Don B. (1999). *ECC, Future resiliency and High Security System.* http://www.certicom.com Oktober 2004.
6. Johnson, Don. Menezes, A. Vanstone, S. (2000). *The Elliptic Curve Digital Signature Algorithm (ECDSA).* http://www.certicom.com, Desember 2004.
7. Jurisic, A. Menezes, A. (2000). *Elliptic Curve and Cryptography.* http://citeseer.ist.psu.edu/cache/papers/ Agustus 2004.
8. Koblitz, N. Menezes, A. Vanstone, S. (2000). *The State of Elliptic Curve Cryptography.* http://www.cacr.math.uwaterloo.ca, Desember 2004.
9. Lopez, Julio. Dahab, Ricardo. (2000). *An Overview of Elliptic Curve Cryptography*. http://citeseer.ist.psu.edu/cache/papers/, Agustus 2004.
10. Menezes,A. van Oorschot, P. Vanstone, S. (1997). *Handbook of Applied Cryptography.* CRCPress.http://www.cacr.math.uwaterloo.ca/hac/., Oktober 2004
11. Munir, Rinaldi.(2003). *Kumpulan Bahan Kuliah Kriptografi*.http://www.mail.informatika.org/~rinaldi/, Oktober 2004
12. Robshaw, JB. Lisa Yin, Yiquin . (1997). *Elliptic Curve Cryptosystem*. http://RSASecurity.com , Agustus 2004.
13. Schneier, Bruce.(1996). *Applied Cryptography, second edition*.John Wiley & Sons, inc.
14. Zuccherato, Robert.(2000). *Elliptic Curve Cryptography Support in Entrust*. http://www.entrust.com , Oktober 2004.